DIGITAL WALLET SIMULATOR

Project Documentation

---

1. Project Title

Digital Wallet Simulator

---

2. Introduction

The **Digital Wallet Simulator** is a menu-driven command-line application developed using **Core Java, JDBC, and MySQL**.
It simulates the basic functionality of a digital wallet where users can create a wallet, add money, withdraw money, and check the wallet balance.

This project is designed to demonstrate:

- Java programming concepts
- JDBC database connectivity
- Layered architecture
- CRUD operations using MySQL

---

3. Objectives

- To understand **JDBC and database connectivity**
- To implement **DAO and Service layers**
- To perform **Create, Read, Update operations**
- To simulate real-world wallet transactions
- To build a clean, modular Java application

---

4. Technologies Used

Technology	Description
Java	Core Java (JDK 8+)
JDBC	Database connectivity
MySQL	Relational Database
VSCode	
Git & GitHub	Version Control

---

5. System Architecture

The project follows a **Layered Architecture**:

Main Layer

↓

Service Layer

↓

DAO Layer

↓

Database (MySQL)

---

6. Project Structure

```
DigitalWalletSimulator/
│
├── src/
│   ├── main/
│   │   └── WalletApp.java
│   │
│   ├── dto/
│   │   └── Wallet.java
│   │
│   ├── dao/
│   │   └── WalletDAO.java
│   │
│   ├── daoimpl/
│   │   └── WalletDAOImpl.java
│   │
│   ├── service/
│   │   └── WalletService.java
│   │
│   └── util/
│       └── DBConnection.java
│
├── database/
│   └── digital_wallet.sql
│
├── README.md
```

└── mysql-connector-j.jar

---

7. Module Description

7.1 DTO Layer (Wallet.java)

- Represents wallet data
- Contains fields like:
    - walletId
    - userName
    - balance
- Used to transfer data between layers

7.2 DAO Layer

WalletDAO (Interface)

- Declares database operations:
    - createWallet()
    - addMoney()
    - withdrawMoney()
    - getBalance()

WalletDAOImpl (Implementation)

- Contains JDBC logic
- Executes SQL queries
- Interacts directly with MySQL

7.3 Service Layer (WalletService.java)

- Acts as a bridge between Main and DAO
- Contains business logic

- Calls DAO methods

7.4 Utility Layer (DBConnection.java)

- Establishes connection with MySQL database
- Loads MySQL JDBC Driver
- Returns Connection object

7.5 Main Layer (WalletApp.java)

- Menu-driven program
- Accepts user input
- Calls service methods
- Displays output to user

---

8. Database Design

Database Name

digital_wallet

Table: wallet

Column Name	Data Type
wallet_id	VARCHAR(20) (PK)
user_name	VARCHAR(50)
balance	DOUBLE

---

9. Functionalities

1. **Create Wallet**
    - Creates a new wallet with initial balance
2. **Add Money**

- Adds amount to existing wallet

3. **Withdraw Money**
    - Withdraws amount if sufficient balance exists
4. **View Balance**
    - Displays current wallet balance
5. **Exit**
    - Terminates application

---

10. Sample Menu Output

==== DIGITAL WALLET SIMULATOR ====

1. Create Wallet

2. Add Money

3. Withdraw Money

4. View Balance

5. Exit

Enter choice:

---

11. Error Handling

- Handles SQL Exceptions
- Validates insufficient balance during withdrawal
- Prevents null database connections

---

12. Advantages

- Simple and easy to understand
- Follows clean architecture
- Real database usage

- Ideal for beginners and academic projects

---

13. Limitations

- Console-based UI
- Single-user system
- No authentication

---

14. Future Enhancements

- GUI using JavaFX / Swing
- User authentication
- Transaction history
- REST API version
- Mobile or web integration

---

15. Conclusion

The **Digital Wallet Simulator** successfully demonstrates how Java applications interact with databases using JDBC. It helps in understanding layered architecture, clean coding practices, and real-world transaction handling. This project is suitable for academic submissions, GitHub portfolios, and interview discussions.